APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/079,928	02/19/2002	Martin D. Richek	3434-P02437US1	6502

110   7590   04/21/2006

DANN, DORFMAN, HERRELL & SKILLMAN
1601 MARKET STREET
SUITE 2400
PHILADELPHIA, PA   19103-2307

EXAMINER
RAMPURIA, SATISH

ART UNIT	PAPER NUMBER
2191	

DATE MAILED: 04/21/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	10/079,928	RICHEK, MARTIN D.
	Examiner	Art Unit	
	Satish S. Rampuria	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>08 February 2006</u>.

2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-35* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-35* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)
2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5)☐ Notice of Informal Patent Application (PTO-152)
6)☐ Other: _____.

Response to Amendment

1.   This action is in response to the Amendment received on Feb 08, 2006.

2.   Claims pending in the application: 1-35.

Response to Arguments

3.   Applicant's arguments with respect to claims 1, 3, 13, 23 and 35 have been

considered but they are not persuasive.

In the remarks, the applicant has argued that:

(i)   The three smart pointers 14, 16 and 18 of George point to a single object.

Pointers that point to a single object are not, and cannot disclose or suggest,

pointers that point to an object of a first class and a second class. There is

only one object in George to which the three smart pointers 14, 16 and 18

point. A single object is not inherently an object of a first class an object of a

second class. Thus, the cited text of George fails to disclose Applicant's

claimed features of "an object of a first class" and "an object of a second

class" as recited in claim 1. Consequently, there is no need or suggestion to

provide function for automatically converting a smart pointer to an object of a

first class to a smart pointer to an object of a second class," since all three

smart pointers 14, 16 and 18 point to a single object. The mere existence of

more than one smart pointer does not automatically lead to the conclusion

that the cited text inherently discloses Applicant's claimed feature of

"providing a function for automatically converting a smart pointer to an object

of a first class to a smart pointer to an object of a second class." Thus, the

assertion in the Office Action that the "conversion function is inherent to

George system since the system has three smart pointers" is misplaced, and

the cited text in George fails to disclose implicitly or inherently Applicant's

claimed feature of a providing a function for automatically converting a smart

pointer to an object of a first class to a smart pointer to an object of a second

class."

(ii)   Oliver clearly fails to disclose al least Applicant's claimed feature of,

"comparing the value of the next pointer to the value of the memory location

of the smart pointer which the selected next pointer is included" as recited in

claims 3, 13, 23 and 35.

Examiner's response:

(i)   In response to applicant's arguments against the references individually, one

cannot show nonobviousness by attacking references individually where the

rejections are based on combinations of references. See *In re Keller*, 642

F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091,

231 USPQ 375 (Fed. Cir. 1986). It is noted that the rejection clearly points out

the where the combination of Oliver and George teach the claimed features

and why it would have been obvious to combine their teachings. Specifically,

the rejection points out that the motivation to "providing a function for

automatically converting a smart pointer to an object of a first class to a smart

pointer to an object of a second class" would be to provide an enhanced and efficiently linking objects in object oriented environment. Further, George specifically discloses that three smart pointer are pointing to the attributes and since the location of the attributes values in the memory pool are typically random, each of the three identified spaces is associated with a node of object list (col. 6,I lines 54-60). Applicants only make general allegation. Therefore, the rejection is proper and maintained herein.

(ii)   In response to Applicants argument, as described previously Oliver discloses the method and apparatus to perform memory management in an object-oriented programming. Specifically, for the limitation comparing...value... pointer, Oliver discloses the pointers next and previous are examined (compared) before they are deleted from the memory (see col. 5, lines 30-46 and Fig. 5D step 502 and related text) (Emphasis added). Therefore, the rejection is proper and maintained herein.

4.   **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later

than SIX MONTHS from the mailing date of this final action.

Claim Rejections - 35 USC § 103

5.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section
102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the
subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary
skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the
invention was made.

6.   Claims 1 and 2 rejected under 35 U.S.C. 103(a) as being unpatentable over US

Patent No. 6,144,965 to Oliver (hereinafter called Oliver) in view of US Patent No.

6,223,342 to George (hereinafter, George).

Per claims 1:

Oliver disclose:

-   A computer-implemented method for performing memory management of an

    object in an object-oriented programming environment using smart pointers (col.

    2, lines "a method and apparatus for performing memory management in an

    object oriented programming environment"), comprising the steps of:

-   providing a base class common to all types of smart pointers (col. 2, lines 7-8

    "base class creates a reference counter for smart pointer to the object");

- providing at least one smart pointer comprising an object pointer for pointing to an object (col. 4, lines 38-39 "a single reference pointer pointing to he original object"), the smart pointer having a next link for pointing to a subsequent smart pointer on a ring and a previous link for pointing to a previous smart pointer on the ring (col. 5, lines 6-8 "The pointer include a standard pointer to the object, a "next pointer" pointer and a "previous pointer" pointer"); and

- wherein the first class and the second class share the common base class (col. 5, lines 12-22 "second entry... linked to each other" also, see FIG. 2A and 5B and related discussion).

Oliver does not explicitly disclose providing a function for automatically converting a smart pointer to an object of a first class to a smart pointer to an object of a second class.

However, George discloses in an analogous computer system providing a function for automatically converting a smart pointer to an object of a first class to a smart pointer to an object of a second class (col. 6, lines 40-60 "The three smart pointers... of the object list 10 contain identifications of memory space for the three attributes of the object. FIG. 2 is a portion of a memory pool 42 of objects and their attribute values. The list element 20 of smart pointers 14, 16 and 18 and the local configuration stream 12 is shown as being stored in the memory pool 42. The properties of each smart pointer are the memory pool index and the offset into the specified memory pool... the smart pointer 14 may identify the memory pool... second

smart pointer 16 also indexes the memory pool 42, but identifies the space 46 as the

location that stores the integer... third smart pointer 18 identifies the space 48 in

memory pool 42 as the location of the 35-byte string of the third attribute"). The

conversion function is inherent to George system since the system has the three smart

pointers.

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to incorporate the method of providing a function for

automatically converting a smart pointer to an object of a first class to a smart pointer

to an object of a second class as taught by George into the method of performing

memory management in an object-oriented environment using smart pointers as

taught by Oliver. The modification would be obvious because of one of ordinary skill in

the art would be motivated to have an object of fist class an object of second class and

the smart pointers to point to the first and second class (objects as class term used by

George) to provide an enhanced and efficiently linking objects in object oriented

environment (linked list) as suggested by George (col. 2, lines 25-35).

Per claim 2:

The rejection of claim 1 is incorporated, and further, Oliver disclose:

- providing single member test for determining if a selected smart pointer is the

   only member of the ring and providing a deletion means for deleting the object if

   the selected smart pointer is determined to be the only member of the ring (col.

   5, lines 23-29 "To delete a pointer... unreferenced... be deleted").

Claim Rejections - 35 USC § 102

7.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that

form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless --
(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in
the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent
by another filed in the United States before the invention by the applicant for patent, except that an international
application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an
application filed in the United States only if the international application designated the United States and was
published under Article 21(2) of such treaty in the English language.

8.   Claim 3-35 are rejected under 35 U.S.C. 102(e) as being anticipated by US

Patent No. 6,144,965 to Oliver (hereinafter called Oliver).

Per claim 3:

Oliver disclose:

-   providing a smart pointer for association with a memory-resident element (col. 5,

    lines 12-13 "second entry in the pointer list is created for the same object"), the

    smart pointer including a next pointer (col. 5, lines 15-16 ""the second pointer...

    includes a "next pointer""");

-   providing an assignment means for assigning the next pointer to point to the

    smart pointer thereby creating a linked list comprising the smart pointer (col. 5,

    lines 17-20 "the "next pointer"... linked to each other"); and

-   providing a comparison means for comparing the value of the next pointer to the

    value of the memory location of the smart pointer in which the selected next

    pointer is included, whereby a determination can be made if the ring contains

    more than one smart pointer (col. 5, lines 34-37 "FIG. 5D... Prior to deleting a

    pointer, the "next pointer" and "previous pointer" pointers are examined

(compared)... If the "next pointer" pointer is the same as the "previous pointer" then there is clearly only one pointer remaining in the list... final pointer is deleted, no pointers will remain in the list. Thus, if the "next pointer" pointer is the same as the "previous pointer" pointer, then the object... deleted... and the last pointer to the object... deleted... If the "next pointer" pointer is not the same as the "previous pointer"... other pointers remaining that point to the object. In this case, a pointer is removed from the list in step 508 and then the pointer is deleted in step 506").

Per claim 4:

The rejection of claim 3 is incorporated, and further, Oliver disclose:

- wherein the method comprises the step of providing a common base to the smart pointer (col. 2, lines 7-8 "base class creates a reference counter for smart pointer to the object").

Per claim 5:

The rejection of claim 3 is incorporated, and further, Oliver disclose:

- wherein the element is an object in an object-oriented programming environment (col. 3, lines 28-30 "a method... in an object-oriented programming environment").

Per claims 6:

The rejection of claim 5 is incorporated, and further, Oliver disclose:

- wherein the smart pointer includes an object pointer for pointing to the object
  (col. 4, lines 42-44 "a copy of the original reference pointer is mad, the new
  reference pointer also points to the original object and its associated count
  object").

Per claims 7, 10, 17, 20, 25, and 28:

The rejection of claim 3 is incorporated, and further, Oliver disclose:

- wherein the linked list comprises a ring (see Fig. 5C and related discussion).

Per claim 8:

Oliver disclose:

- wherein the smartpointer includes a previous pointer. The limitations in the
  claims are similar to those in claim 3, and rejected under the same rational set
  forth in connection with the rejection of claim 3.

Per claims 9, 19, and 27:

The rejection of claim 8 is incorporated, and further, Oliver disclose:

- providing an assignment means for assigning the previous pointer to point to the
  smart pointer, thereby creating a bi-directional, doubly-linked list (see Fig. 5B and
  related discussion).

Per claims 11, 14, 21, 24, and 29:

The rejection of claim 3 is incorporated, and further, Oliver disclose:

- providing a deletion means for deleting the memory-resident element associated
with the smart pointer (col. 5, lines 22-23 "delete a pointer... is deleted") if the
value of the next pointer of the smart pointer is equal to the value of the memory
location of the smart pointer in which the next pointer is included (col. 4, lines 55-
67 "each time a pointer is deleted... deleted in step 408").

Per claims 12, 22, and 30:

The rejection of claim 3 is incorporated, and further, Oliver disclose:

- wherein the smart pointer includes a first smart pointer, and wherein the method
comprises the step of providing an attachment means for attaching a second
smart pointer associated with the memory-resident element to the linked list
element (col. 5, lines 12-20 "second entry in the pointer list is created for the
same object... linked to each other").

Per claim 13:

Oliver disclose:

- providing a linked list comprising a smart pointer associated with a memory-
resident element (col. 5, lines 12-13 "second entry in the pointer list is created for
the same object"), the smart pointer including a next-pointer for pointing to the
smart pointer (col. 5, lines 17-20 "the "next pointer"... linked to each other"); and

- providing a comparison means for comparing the value of memory of the smart

  pointer to the value of the next pointer of the smart pointer, to provide whether

  the linked list contains only the smart pointer (col. 5, lines 34-37 "pointer...

  examined... pointer is the same... one pointer remaining in the list").

Per claim 15:

The rejection of claim 13 is incorporated, and further, Oliver disclose:

- wherein the element is an object in an object-oriented programming environment.

  The limitations in the claims are similar to those in claim 23, and rejected under

  the same rational set forth in connection with the rejection of claim 23.

Per claim 16:

- wherein the smart pointer includes an object pointer for pointing to the object.

  The limitations in the claims are similar to those in claim 13, and rejected under

  the same rational set forth in connection with the rejection of claim 13.

Per claim 18:

- wherein the smart pointer includes a previous pointer. The limitations in the

  claims are similar to those in claim 13, and rejected under the same rational set

  forth in connection with the rejection of claim 13.

Per claims 23, 35:

- providing a linked list comprising a first smart pointer and a second smart pointer each associated with a memory-resident element (col. 5, lines 12-13 "second entry in the pointer list is created for the same object"), the first smart pointer including a first next-pointer for pointing to the second smart pointer and the second smart pointer including a second next-pointer for pointing to the first smart pointer (col. 5, lines 15-16 "the second pointer... includes a "next pointer""); and

- providing a comparison means for comparing the value of the memory location of a selected smart pointer giving up its association with the memory-resident element to the value of the next-pointer of the selected smart pointer (col. 5, lines 17-20 "the "next pointer"... linked to each other"), to provide a determination whether the linked list contains only the selected smart pointer (col. 5, lines 34-37 "pointer... examined... pointer is the same... one pointer remaining in the list").

Per claim 26:

- wherein the first smart pointer and the second smart pointer each include a previous pointer. The limitations in the claims are similar to those in claim 23, and rejected under the same rational set forth in connection with the rejection of claim 23.

Per claim 31:

- comprising the step of providing a common base to the smart pointers. The

  limitations in the claims are similar to those in claim 23, and rejected under the

  same rational set forth in connection with the rejection of claim 23.

Per claim 32:

The rejection of claim 23 is incorporated, and further, Oliver disclose:

- wherein the element is an object in an object-oriented programming environment.

  The limitations in the claims are similar to those in claim 23, and rejected under

  the same rational set forth in connection with the rejection of claim 23.

Per claim 33:

The rejection of claim 32 is incorporated, and further, Oliver disclose:

- wherein the first smart pointer and the second smart pointer each include an

  object pointer for pointing to the object. The limitations in the claims are similar to

  those in claim 23, and rejected under the same rational set forth in connection

  with the rejection of claim 23.

Per claim 34:

The rejection of claim 32 is incorporated, and further, Oliver disclose:

- wherein the first smart pointer is associated with a first object of a first class and

  the second smart pointer is associated with a second object of a second class,

  and wherein the method comprises the step of providing a conversion means for

providing automatic conversion between the first smart pointer and the second

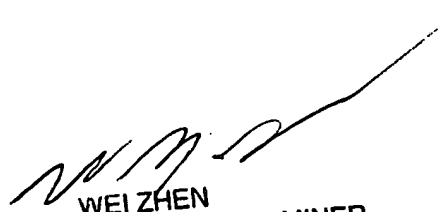smart pointer (col. 5, lines 12-22 "second entry... linked to each other").

Conclusion

9.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to **Satish S. Rampuria** whose telephone number is **(571)**

272-3732. The examiner can normally be reached on **8:30 am to 5:00 pm** Monday to

Friday except every other Friday and federal holidays. Any inquiry of a general nature or

relating to the status of this application should be directed to the **TC 2100 Group**

receptionist: 571-272-2100

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, **Wei Y. Zhen** can be reached on **(571) 272-3708**. The fax phone number for

the organization where this application or proceeding is assigned is **571-273-8300**.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free).

Satish S. Rampuria
Patent Examiner/Software Engineer
Art Unit 2191

WEI ZHEN
SUPERVISORY PATENT EXAMINER